Exercise Worksheet

www.vomtom.at

From the Course:
Understanding Docker Run, Dockerfile, Docker-Compose for Beginners

Using networks in docker-compose for better segregation

Let’s use this docker-compose.yml file:

```
version: "3.7"

services:
  web:
    image: nginx:alpine
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
    ports:
      - 8080:80
    networks:
      - app1_net
      - app2_net

  app1:
    image: httpd:latest
    networks:
      - app1_net

  app2:
    image: httpd:latest
    networks:
      - app2_net

networks:
  app1_net:
  app2_net:
```

And the following nginx.conf configuration file in the same directory:

```
events {}
http {
    server {
    listen 80;
    listen [::]:80;

    server_name example.com;

    location / {
        proxy_pass http://app1:80/;
    }
    location /app2 {
        proxy_pass http://app2:80/;
    }
  }
}
```

Now go to the terminal and type in:

```
docker-compose up
```

And wait for the services to come up and show the log-output.

Go to http://localhost:8080 and observer the command line

- It will show you the nignx-container web_1 container (reverse_proxy) was requested
- And forwarded the request to “app1” container
- Reload a few times to make this more obvious

Go to http://localhost:8080/app2 and observe the command line

- It will show you again that nginx-container web_1 container (reverse_proxy) was requested
- And now forwards to “app2” container
- Reload a few times to make this more obvious